

# Lettre d'information des IA-IPR de mathématiques – Septembre 2017

## L'algorithmique et la programmation au lycée

En cette rentrée scolaire, l'algorithmique et la programmation deviennent l'une des quatre parties du [programme de mathématiques de la classe de seconde](#), ce qui implique des évolutions dans les pratiques d'enseignement. Cette lettre présente quelques informations susceptibles d'éclairer ces changements.

### 1. Le choix de l'outil de programmation

Le programme précise que « le choix du langage se fera parmi les langages interprétés, concis, largement répandus, et pouvant fonctionner dans une diversité d'environnements ». Comme cela a été présenté dans les formations mises en œuvre à la fin de la précédente année scolaire, le langage *Python* répond parfaitement à ces objectifs. Il est donc essentiel qu'il puisse être utilisé dans tous les lycées.

Pour l'installation, le choix pourra par exemple se porter sur [EduPython](#), qui est une distribution clé en main, complète et portable pour programmer sous un environnement *Python 3*. D'autres exemples de distribution, ainsi que des tutoriels pour apprendre à utiliser le langage *Python*, sont proposés sur la page du site académique [Débuter avec le langage Python](#).

### 2. La continuité entre le collège et le lycée

L'algorithmique et la programmation constituent une part substantielle des [programmes de collège](#). Les collégiens ont donc pratiqué des activités sur ce thème, notamment avec le logiciel [Scratch](#). Ce logiciel peut tout à fait être utilisé par les élèves de seconde en début d'année pour faire la transition avec *Python*. Plus généralement, il est essentiel de s'appuyer sur les acquis des élèves de collège pour aborder ce thème en lycée.

### 3. Ressources pédagogiques

Le site Eduscol a publié un [document ressource sur l'enseignement de l'algorithmique et de la programmation en lycée](#). Il présente notamment le langage *Python* ainsi que des exemples d'activités en classe.

Des exemples d'activités pour la classe sont d'ores et déjà présents sur [la page dédiée du site académique](#), avec notamment un exemple d'évaluation diagnostique et des activités de transition de *Scratch* à *Python*. Cette page sera rapidement et progressivement enrichie dans les mois qui viennent.

#### 4. Formations

Comme indiqué dans la lettre de rentrée, [l'offre de formation en mathématiques](#) comporte deux stages de deux jours chacun.

Le premier stage, [Algorithmique et programmation au lycée](#), se déroulera entre novembre et janvier et s'adresse à des enseignants débutants ou ayant des connaissances modestes sur le langage *Python*. Il permettra de parcourir l'ensemble des notions en lien avec le programme de seconde, à un niveau simple et accessible à tous.

Le second stage, [Programmation avancée en Python](#), se déroulera après le stage précédent, entre janvier et avril. Il s'adresse à des enseignants possédant déjà quelques connaissances de base en *Python* et souhaitant les approfondir.

Il est tout à fait possible de participer aux deux stages. Suite aux nombreuses candidatures, leurs capacités d'accueil vont être augmentées pour répondre à l'ensemble des demandes. Les inscriptions doivent se faire impérativement avant le 29 septembre.

#### 5. L'évolution des sujets du baccalauréat

Dans un objectif de simplicité et de cohérence, il est proposé dès à présent de faire évoluer l'écriture des algorithmes dans les sujets de baccalauréat, conformément aux principes suivants :

- suppression de la déclaration des variables, les hypothèses faites sur les variables étant précisées par ailleurs ;
- suppression des entrées-sorties ;
- simplification de la syntaxe, avec le symbole  $\leftarrow$  pour l'affectation.

Les exemples qui suivent illustrent ces principes sur des sujets de baccalauréat de la session 2017 et montrent qu'ils facilitent la lecture sans changement de fond sur l'algorithmique.

<p><b>BAC ES 2017</b></p> <p>Recopier et compléter l'algorithme de façon qu'il affiche le montant total des cotisations de l'année 2017.</p> <p>Variables      S est un nombre réel                     N est un entier                     U est nombre réel</p> <p>Initialisation    S prend la valeur 0                     U prend la valeur 900                     Pour N allant de 1 à 12 :                         Affecter à S la valeur ...                         Affecter à U la valeur <math>0,75 U + 12</math>                     Fin Pour</p>	<p><i>On propose simplement un changement de forme : suppression des étiquettes « Variables » et « Initialisation », suppression de la déclaration des variables, remplacement de la syntaxe d'une affectation.</i></p> <p style="text-align: right;"> <math>S \leftarrow 0</math>  <math>U \leftarrow 900</math>            Pour N allant de 1 à 12  <math>S \leftarrow \dots</math>  <math>U \leftarrow 0,75 U + 12</math>            Fin Pour         </p>
<p><b>BAC S 2017</b></p> <p>On considère l'algorithme suivant :</p> <p>Variables      <math>\lambda</math> est un réel positif                     S est un réel strictement compris entre 0 et 1</p> <p>Initialisation    Saisir S                     <math>\lambda</math> prend la valeur 0</p> <p>Traitement      Tant que <math>1 - \frac{\lambda+1}{e^\lambda} &lt; S</math> faire                         <math>\lambda</math> prend la valeur <math>\lambda + 1</math>                     Fin Tant que</p> <p>Sortie            Afficher <math>\lambda</math></p> <p>a. Quelle valeur affiche cet algorithme si on saisit la valeur <math>S = 0,8</math> ? b. Quel est le rôle de cet algorithme ?</p>	<p><i>On propose de supprimer la déclaration des variables, mais que l'énoncé précise les hypothèses faites sur les variables, de simplifier la syntaxe, de renoncer aux entrées sorties.</i></p> <p>On considère l'algorithme suivant, où la variable S désigne un réel de l'intervalle ]0,1[.</p> <p style="text-align: right;"> <math>\lambda \leftarrow 0</math>            Tant que <math>1 - \frac{\lambda+1}{e^\lambda} &lt; S</math> faire  <math>\lambda \leftarrow \lambda + 1</math>            Fin Tant que         </p> <p>a. Si la variable S contient la valeur 0,8 avant l'exécution de cet algorithme, que contient la variable <math>\lambda</math> à la fin de son exécution ? b. Quel est le rôle de cet algorithme ?</p>
<p><b>BAC STI2D 2017</b></p> <p>Voici un algorithme qui, lorsque l'on saisit un nombre N non nul de jours écoulés, calcule et affiche la masse de gaz restant dans le système.</p> <p>Variables      N : un nombre entier naturel                     k : un nombre entier naturel                     u : un nombre réel</p> <p>Entrée            Saisir N</p> <p>Initialisation    u prend la valeur 660</p> <p>Traitement      Pour k allant de 1 à ...                         u prend la valeur ...                     Fin pour</p> <p>Sortie            Afficher u</p> <p>Recopier et compléter la partie relative au traitement de cet algorithme.</p>	<p><i>On propose la suppression de la déclaration de variables et des entrées-sorties, la simplification de la syntaxe.</i></p> <p>Voici un algorithme qui calcule la masse u de gaz restant dans le système après un nombre entier strictement positif N de jours écoulés.</p> <p style="text-align: right;"> <math>u \leftarrow 660</math>            Pour k allant de 1 à ...  <math>u \leftarrow \dots</math>            Fin pour         </p> <p>Recopier et compléter cet algorithme.</p>

<p><b>BAC STLbio 2017</b></p> <p>Soit l'algorithme suivant :</p> <p>Variables     <math>n</math> entier naturel                   <math>C</math> réel</p> <p>Initialisation  Affecter à <math>n</math> la valeur 0                   Affecter à <math>C</math> la valeur 3,4</p> <p>Traitement     Tant que <math>C</math> est supérieur à 1                   Affecter à <math>n</math> la valeur <math>n+1</math>                   Affecter à <math>C</math> la valeur <math>0,8 \times C</math>                   Fin tant que</p> <p>Sortie           Afficher <math>n</math></p> <p>Quelle valeur affiche l'algorithme ? Interpréter le résultat dans le contexte de cet exercice.</p>	<p><i>On propose la suppression de la déclaration de variables et des entrées-sorties, la simplification de la syntaxe.</i></p> <p>Soit l'algorithme suivant :</p> <p style="padding-left: 40px;"><math>n \leftarrow 0</math> <math>C \leftarrow 3,4</math> Tant que <math>C \geq 1</math>           <math>n \leftarrow n + 1</math>           <math>C \leftarrow 0,8 \times C</math> Fin Tant que</p> <p>Quelle est la valeur de la variable <math>n</math> à la fin de l'exécution de l'algorithme ? Interpréter le résultat dans le contexte de l'exercice.</p>
<p><b>BAC STMG 2017</b></p> <p>On considère l'algorithme suivant :</p> <p>Variables     <math>n</math> est un nombre entier                   <math>u</math> et <math>k</math> sont des nombres réels</p> <p>Traitement     Saisir <math>k</math>                   <math>n</math> prend la valeur 0                   <math>u</math> prend la valeur 3 081,45                   Tant que <math>u &lt; k</math> Faire                   <math>u</math> prend la valeur <math>1,04 \times u</math>                   <math>n</math> prend la valeur <math>n + 1</math>                   Fin Tant que                   Afficher <math>n</math></p> <p>Si l'on choisit <math>k = 4\ 000</math>, quelle valeur affichera cet algorithme ? Interpréter ce résultat dans le contexte étudié.</p>	<p><i>On propose la suppression de la déclaration de variables et des entrées-sorties, la simplification de la syntaxe. Par cohérence d'un sujet à l'autre on propose de ne garder que : Tant que ... plutôt que : Tant que ... faire</i></p> <p style="padding-left: 40px;"><math>n \leftarrow 0</math> <math>u \leftarrow 3081,45</math> Tant que <math>u &lt; k</math>           <math>u \leftarrow 1,04 \times u</math>           <math>n \leftarrow n + 1</math> Fin Tant que</p> <p>Quelle est la valeur de la variable <math>n</math> à la fin de l'exécution de cet algorithme si la valeur de la variable <math>k</math> en début d'exécution est égale à 4000 ? Interpréter ce résultat dans le contexte étudié.</p>